

Comprehensive Notes On Database Management System

Using LibreOffice Base

Class X - Information Technology (Subject Code 402)

Contents

1	Introduction to Database Management System (DBMS)	3
1.1	Data vs. Information	3
1.2	Databases and Database Management Systems (DBMS)	3
1.3	Key Advantages of DBMS over Traditional File Systems	4
1.4	Database Models	5
1.5	Relational Database Concepts and Terminology	5
1.6	Objects of an RDBMS	5
2	Creating Database and Tables in LibreOffice Base	6
2.1	Launching LibreOffice Base and the Database Wizard	6
2.2	Components of LibreOffice Base Interface	6
2.3	Methods to Create Tables in LibreOffice Base	7
2.3.1	Method 1: Creating Table Using Wizard	7
2.3.2	Method 2: Creating Table in Design View	7
2.4	Data Types in LibreOffice Base	8
2.5	Data Entry, Editing, Sorting, and Searching in Datasheet View	9
3	Editing Tables and Setting Relationships	10
3.1	Modifying Table Structure	10
3.2	Table Management Operations	10
3.3	Data Redundancy, Inconsistency, and Relationships	11
3.4	Types of Relationships between Tables	11
3.5	Referential Integrity	11
3.5.1	Referential Integrity Update and Delete Rules	11
3.6	Creating Relationships in LibreOffice Base	12
4	Retrieving Data using Query	12
4.1	Methods to Create a Query in Base	12
4.2	Structure of the Query Design Window	13
4.3	Applying Filter Conditions and Operators	13
4.4	Calculations, Grouping, and Aggregate Functions	14
4.5	Direct SQL Execution in Base	14
5	Creating Forms and Reports	14
5.1	Forms: Concept and Benefits	15
5.2	Creating Forms Using Form Wizard	15
5.3	Form Controls and Properties	15
5.4	Reports: Concept and Features	16
5.5	Creating Reports Using Report Wizard	16
6	Comprehensive Question Bank	17
6.1	Multiple Choice Questions (MCQs)	17
6.2	Assertion-Reasoning Questions	19
6.3	One-Word / Brief Answer Questions	20
6.4	Differentiate Between Questions	21
6.5	Short Answer Questions	22
6.6	Long Answer Questions	22

6.7 Case-Study / Competency-Based Questions	23
---	----

1 Introduction to Database Management System (DBMS)

In modern society, managing data efficiently is of paramount importance for any enterprise, educational institution, government body, or healthcare organization. A Database Management System (DBMS) serves as the core infrastructure responsible for entering, storing, modifying, and retrieving vast amounts of information securely and quickly.

1.1 Data vs. Information

- **Data:** Data refers to raw, unorganized facts, figures, symbols, or observations recorded from real-world entities. Data in its raw state lacks actionable meaning or logical context.
 - *Relevance:* Serves as the basic input for computing and analytical processing.
 - *Examples:* Scores such as 95, 88, 72; names like "Rohan", "Ananya"; product quantities like 15, 100.
- **Information:** Information is the output obtained when raw data is collected, structured, processed, contextualized, and presented in a meaningful format.
 - *Relevance:* Enables human decision-making, analysis, and communication.
 - *Examples:* "Rohan scored 95 in Science", a complete student report card, an inventory stock sheet, or a salary pay slip.

Table 1: Comparative Analysis: Data vs. Information

Parameter	Data	Information
Definition	Unprocessed, raw facts and figures.	Processed and structured data with context.
Nature	Atomic and unorganized.	Meaningful and organized.
Dependency	Independent; does not rely on information.	Dependent entirely on data inputs.
Usage	Used directly as input for computer systems.	Used by humans for decision making.
Example	List of numbers: 45, 92, 78.	"Average test mark of Class X is 71.6%".

💡 Important Note

NCERT Key Observation: Raw facts alone cannot answer questions. For instance, the number 38.5 is raw data. When contextualized as "Body Temperature = 38.5°C", it becomes vital clinical information indicating fever.

1.2 Databases and Database Management Systems (DBMS)

- **Database:** A database is an organized collection of logically related data items stored electronically in a central repository. It is designed to be easily created, accessed, updated, managed, and queried.

- *Analogies:* A telephone directory, a library card catalog, a school student admission register, a hospital patient registry.
- **Database Management System (DBMS):** A DBMS is a specialized software application that acts as an interface between end-users (or application software) and the underlying database file. It provides administrative tools to define, construct, manipulate, and control database access.
 - *Popular Software Examples:* LibreOffice Base (Open Source), MS Access, MySQL, Oracle Database, PostgreSQL, IBM DB2, Microsoft SQL Server.

1.3 Key Advantages of DBMS over Traditional File Systems

Using a DBMS provides foundational advantages compared to legacy manual or file-based management systems:

1. **Control of Data Redundancy:** Data redundancy refers to unnecessary duplication of identical data across multiple files. DBMS eliminates redundancy by storing data in a single central repository.
2. **Enforcement of Data Consistency:** By controlling redundancy, DBMS guarantees consistency. Updating an employee's address once in the central table automatically reflects across all departmental views, avoiding conflicting records.
3. **Data Sharing:** Authorized multiple users and applications can simultaneously access and query the same centralized database safely.
4. **Data Security:** Strict access control protocols, authentication mechanisms, and user roles ensure that sensitive fields (e.g., salary, passwords) are restricted to authorized personnel only.
5. **Data Integrity:** DBMS enforces integrity constraints (rules). For example, preventing negative values in age or ensuring a valid date format.
6. **Efficient Data Search and Retrieval:** Indexed tables and structured query processing allow instantaneous retrieval of records from millions of stored entries.
7. **Automated Backup and Disaster Recovery:** Built-in mechanisms automatically create periodic snapshots and transaction logs to recover data in case of power failures or hardware breakdown.

Guideline / Rule

Core Database Axiom: *Redundancy breeds inconsistency.* If duplicate copies of data exist in separate systems, an update to one copy without updating the others causes data inconsistency.

1.4 Database Models

A data model describes the logical structure, relationships, and constraints governing how data is stored and organized inside a database.

1. **Hierarchical Model:** Data is organized in a tree-like parent-child structure. A parent node can have multiple child nodes, but each child node has exactly one parent node.
2. **Network Model:** An extension of the hierarchical model where data is represented as a network graph. A child node can have multiple parent nodes, allowing complex many-to-many relationships.
3. **Relational Model:** Data is stored in two-dimensional tables consisting of rows and columns. Tables are linked to each other using common key attributes. Invented by Dr. E.F. Codd in 1970, it forms the foundation of modern RDBMS systems including LibreOffice Base.

1.5 Relational Database Concepts and Terminology

- **Entity:** Any identifiable real-world object, concept, or event about which data is captured (e.g., Student, Teacher, Book, Order).
- **Table (or Relation):** A 2D grid structure representing an entity, composed of named vertical columns and horizontal rows.
- **Field (or Column / Attribute):** A named vertical column representing a specific property or characteristic of the entity (e.g., RollNo, StudentName, DOB).
- **Record (or Row / Tuple):** A single horizontal row containing a complete set of attribute values for one instance of the entity.
- **Primary Key:** A field or set of fields that uniquely identifies each record in a table. A primary key cannot contain duplicate values and can never be NULL.
- **Composite Primary Key:** A primary key formed by combining two or more attributes when no single attribute uniquely identifies a record.
- **Candidate Key:** Any attribute or combination of attributes eligible to serve as a primary key due to unique non-null values.
- **Alternate Key:** A candidate key that was not chosen as the primary key.
- **Foreign Key:** An attribute in a child table that references the primary key of a parent table, establishing a logical link between the two relations.

1.6 Objects of an RDBMS

LibreOffice Base structures database applications using four major objects:

1. **Tables:** The fundamental building blocks used to store raw records in rows and columns.

2. **Queries:** Saved filtering questions used to extract, calculate, or manipulate specific records matching user criteria across tables.
3. **Forms:** User-friendly graphical front-end screens designed for smooth, guided data entry, editing, and record navigation.
4. **Reports:** Formatted summaries designed for printing, exporting, or presenting processed database insights to decision-makers.

2 Creating Database and Tables in LibreOffice Base

LibreOffice Base is an open-source relational database management system package included in the LibreOffice suite. Base saves complete database projects into a single file with the extension `.odb` (OpenDocument Database).

2.1 Launching LibreOffice Base and the Database Wizard

To start LibreOffice Base:

1. Click **Start** → **LibreOffice** → **LibreOffice Base**.
2. The **Database Wizard** dialog box appears, presenting options:
 - *Create a new database (Default).*
 - *Open an existing database file.*
 - *Connect to an existing external database (e.g., MySQL, Oracle).*
3. Select **Create a new database** and click **Next**.
4. Choose whether to register the database in LibreOffice and open it for editing, then click **Finish**.
5. Save the database file with a descriptive name (e.g., `SchoolManagement.odb`).

2.2 Components of LibreOffice Base Interface

- **Title Bar:** Displays the active database filename (e.g., `SchoolManagement.odb`) and application title.
- **Menu Bar:** Contains main menu drop-downs (File, Edit, View, Insert, Tools, Window, Help).
- **Standard Toolbar:** Provides quick-action icons for saving, editing, cutting, copying, pasting, and table design operations.
- **Database Objects Pane (Left Panel):** Allows users to switch between the four primary objects: **Tables**, **Queries**, **Forms**, and **Reports**.

- **Tasks Pane (Top Right Panel):** Lists available creation methods for the selected object (e.g., "Create Table in Design View...", "Use Wizard to Create Table...").
- **Work Area / Content Pane (Bottom Right):** Displays existing created tables, queries, forms, or reports.
- **Status Bar:** Located at the bottom; displays current database engine connection status (e.g., HSQLDB Embedded).

2.3 Methods to Create Tables in LibreOffice Base

LibreOffice Base offers two primary methods to construct tables:

2.3.1 Method 1: Creating Table Using Wizard

The Table Wizard provides pre-designed templates for business and personal databases:

1. In the Database Objects Pane, click **Tables**.
2. In the Tasks Pane, click **Use Wizard to Create Table....**
3. **Step 1: Select Fields** - Choose a category (*Business* or *Personal*), select a sample table template (e.g., Customers, Employees, Addresses), and move desired fields to the selected fields list using arrow buttons.
4. **Step 2: Set Types and Formats** - Adjust field names, data types, auto-value settings, and maximum length.
5. **Step 3: Set Primary Key** - Check "Create a primary key", choose automatically add an auto-value primary key or specify an existing field.
6. **Step 4: Create Table** - Enter a table name and click **Finish**.

2.3.2 Method 2: Creating Table in Design View

Design View grants total manual control over field architecture:

1. Click **Tables** → **Create Table in Design View....**
2. A two-pane window opens:
 - *Upper Grid Pane:* Columns for **Field Name**, **Field Type**, and **Description**.
 - *Lower Field Properties Pane:* Configuration settings for the selected field (e.g., **Entry required**, **Length**, **Default value**, **Format example**).
3. Type field names and select appropriate data types from the drop-down menu.
4. To set a **Primary Key**: Right-click the gray row selector button beside the chosen field name and select **Primary Key** (a small key icon appears).

5. Click **Save** (or press Ctrl+S), enter a table name (e.g., tbl_Students), and close Design View.

Guideline / Rule

Naming Constraints for Fields in Base:

- Field names can contain letters, numbers, spaces, and underscores.
- Avoid starting field names with numbers or using reserved SQL keywords (e.g., SELECT, WHERE, TABLE).
- Field names should be descriptive (e.g., Student_DOB instead of D).

2.4 Data Types in LibreOffice Base

Data types dictate what kind of values can be stored in a column, memory allocation, and allowed mathematical or logical operations.

Table 2: Comprehensive Classification of LibreOffice Base Data Types

Category	Data Type	SQL Keyword	Description & Capacity
Text Types	Text (fix)	CHAR	Fixed-length string up to set limit. Padding added.
	Text	VARCHAR	Variable-length string up to specified max character limit.
	Text (ignore case)	VARCHAR_IGNORECASE	Case-insensitive text storage and comparisons.
	Memo	LONGVARCHAR	Long descriptive text blocks (paragraphs, notes).
Numeric Types	Tiny Integer	TINYINT	8-bit integer (-128 to 127).
	Small Integer	SMALLINT	16-bit integer (-32,768 to 32,767).
	Integer	INTEGER	32-bit integer (-2,147,483,648 to 2,147,483,647).
	Big Integer	BIGINT	64-bit large integer value.
	Number	NUMERIC	Exact decimal value with set precision & scale.
	Decimal	DECIMAL	Fixed-point money and accounting numbers.
	Real / Float / Double	REAL / DOUBLE	Floating-point numbers with high precision decimals.
Date/Time	Date	DATE	Stores Calendar Date (Year, Month, Day).
	Time	TIME	Stores Clock Time (Hours, Minutes, Seconds).
	Timestamp	TIMESTAMP	Combined Date and Time values.
Binary / Other	Image / Binary	LONGVARBINARY	Stores binary files, pictures, signatures, or documents.
	Yes/No	BOOLEAN	Stores logical True (1) or False (0) flags.

2.5 Data Entry, Editing, Sorting, and Searching in Datasheet View

- **Opening Datasheet View:** Double-click any created table name in the Work Area pane.
- **Entering / Editing Records:** Click inside any cell and type data. Pressing Tab or Enter

moves focus to the next field/row. A pencil icon indicates an active row being edited.

- **Deleting Records:** Select a row by clicking its gray record selector bar on the far left, right-click, and select **Delete Rows**.
- **Sorting Records:** Click on a column header, then click the **Sort Ascending** ($A \rightarrow Z, 0 \rightarrow 9$) or **Sort Descending** ($Z \rightarrow A, 9 \rightarrow 0$) icon on the table toolbar.
- **Record Navigation Bar:** Located at the bottom of Datasheet View; contains buttons to skip to the *First Record*, *Previous Record*, *Next Record*, *Last Record*, or insert a *New Record*.

3 Editing Tables and Setting Relationships

After constructing database tables, administrators frequently need to modify table definitions, establish foreign key connections, and enforce integrity rules.

3.1 Modifying Table Structure

To alter a table's schema after creation:

1. In the Work Area pane, right-click the target table name and choose **Edit**.
2. The table opens in **Design View**.
3. **Adding Fields:** Scroll to a blank row at the bottom of the grid, enter a new field name and select its data type.
4. **Deleting Fields:** Right-click the gray row selector button beside the field to remove and select **Delete**.
5. **Modifying Field Properties:** Click any existing field and alter its properties in the lower pane (e.g., change **Length** or set **Entry required** to **Yes**).
6. Save the table changes.

3.2 Table Management Operations

Right-clicking a table name in Base provides administrative options:

- **Copy:** Creates a duplicate table definition. A popup prompts whether to copy *Definition and data*, *Definition only*, or append data to an existing table.
- **Rename:** Allows changing the table's display name.
- **Delete:** Permanently drops the table and all contained records from the database.

3.3 Data Redundancy, Inconsistency, and Relationships

- **Data Redundancy:** Storing repetitive data across multiple unstructured tables. For instance, recording full customer names and billing addresses on every single sales invoice row.
- **Data Inconsistency:** Occurs when redundant instances of identical data become mismatching due to incomplete updates.
- **Table Relationships:** Links established between primary key columns of a master table and foreign key columns of a child table to minimize redundancy while preserving complete logical record association.

3.4 Types of Relationships between Tables

1. **One-to-One (1:1) Relationship:** Each record in Table A is associated with exactly one record in Table B, and vice versa.
 - *Example:* A `PassportDetails` table linked to a `Citizen` table.
2. **One-to-Many (1:N) Relationship:** A single record in Table A can relate to multiple records in Table B, but a record in Table B relates back to only one record in Table A. This is the most common relationship in RDBMS design.
 - *Example:* One `Department` employs many `Employees`. One `Customer` places many `Orders`.
3. **Many-to-Many (N:M) Relationship:** Multiple records in Table A relate to multiple records in Table B. In relational database implementation, N:M relationships are converted into two 1:N relationships using an intermediate junction table.
 - *Example:* Multiple `Students` enroll in multiple `Courses`.

3.5 Referential Integrity

Referential Integrity is a strict database rule enforcing that foreign key values in a child table must match a valid primary key value in the referenced parent table, or else be explicitly set to NULL.

Important Note

Purpose of Referential Integrity: Prevents orphaned records (e.g., placing an order for a `CustomerID` that does not exist in the `Customers` table) and ensures cascaded consistency when primary key records are modified or deleted.

3.5.1 Referential Integrity Update and Delete Rules

When creating relationships in Base, four options control parent-record updates and deletions:

1. **No Action / Restrict (Default):** Rejects any attempt to delete or update a primary key value in the parent table if related foreign key records exist in the child table.

2. **Update / Delete Cascade:** Automatically updates or deletes all matching foreign key records in the child table whenever the parent primary key is updated or deleted.
3. **Set Null:** Deletes or updates the parent record while automatically resetting matching foreign key values in the child table to NULL.
4. **Set Default:** Resets matching foreign key fields in the child table to a pre-defined default value upon parent record modification.

3.6 Creating Relationships in LibreOffice Base

To set up a relationship between two tables:

1. From the main menu, click **Tools** → **Relationships....**
2. The **Add Tables** dialog box appears. Select the target tables (e.g., `tbl_Customers` and `tbl_Orders`) and click **Add**, then close the dialog.
3. Click and drag the **Primary Key** field name (e.g., `CustomerID`) from the parent table box and drop it directly over the corresponding **Foreign Key** field name in the child table box.
4. A line connecting the two table boxes appears, indicating an active relationship.
5. Double-clicking the relationship connector line opens the ****Relations**** dialog box to configure update/delete cascade options.
6. Click ****Save**** and close the Relationships window.

4 Retrieving Data using Query

A ****Query**** is a database object designed to extract, search, filter, calculate, and compile specific subset records from one or more tables based on user-defined condition criteria.

4.1 Methods to Create a Query in Base

LibreOffice Base provides three methods to construct queries:

1. **Using Wizard to Create Query...** (Guided step-by-step query construction).
2. **Create Query in Design View...** (Visual graphical grid construction; recommended).
3. **Create Query in SQL View...** (Writing raw SQL code statements directly).

4.2 Structure of the Query Design Window

When opening ****Create Query in Design View...****, the window is split into two main sections:

1. **Upper Pane (Tables Pane):** Displays box diagrams of tables/queries added to the query, showing available field lists and relationship join lines.
2. **Lower Pane (Query Design Grid / Criteria Grid):** Contains configurable rows for specifying field actions:
 - **Field:** The name of the field to include in the output.
 - **Alias:** A custom display column header label for the output.
 - **Table:** Specifies the source table for the field.
 - **Sort:** Specifies sorting order (*Ascending*, *Descending*, or *un-sorted*).
 - **Visible:** A check-box determining whether the field is displayed in the final query result.
 - **Criterion:** The exact condition expression used to filter rows (e.g., > 500 , $= \text{'Delhi'}$).
 - **Or:** Additional condition rows for evaluating logical OR expressions.

4.3 Applying Filter Conditions and Operators

Table 3: Common SQL Criteria Operators Used in Base Queries

Operator Type	Syntax / Example	Explanation
Comparison	$= \text{'Science'}$	Matches exact text string.
	> 75	Matches numeric values strictly greater than 75.
	≤ 5000	Matches values less than or equal to 5000.
	$\neq \text{'Mumbai'}$	Matches all records except those with city 'Mumbai'.
String Matching	LIKE 'A%'	Matches strings starting with letter 'A'.
	LIKE '%Singh'	Matches strings ending with 'Singh'.
	LIKE '%Computer%'	Matches strings containing 'Computer' anywhere.
Range / Set	$\text{BETWEEN } 50 \text{ AND } 100$	Matches inclusive numeric range between 50 and 100.
	$\text{IN ('Class X', 'Class XI')}$	Matches records matching any value in list.
Logical	$> 80 \text{ AND Marks } < 90$	Both conditions must evaluate to True.
	$\text{City} = \text{'Delhi'} \text{ OR City} = \text{'Agra'}$	Either condition can evaluate to True.

4.4 Calculations, Grouping, and Aggregate Functions

Queries can summarize data groups using built-in SQL Aggregate Functions:

- `SUM(Field)`: Calculates total sum of numeric values in a column.
- `AVG(Field)`: Calculates average arithmetic mean of numeric values.
- `COUNT(Field)`: Counts total number of non-null records.
- `MIN(Field)`: Returns the minimum lowest value.
- `MAX(Field)`: Returns the maximum highest value.

In Query Design View, selecting **Group** in the **Function** row groups rows by duplicate column values (e.g., grouping sales totals by Region).

4.5 Direct SQL Execution in Base

Users can execute standard Data Manipulation Language (DML) queries directly via **Tools** → **SQL...**:

Listing 1: Standard SQL DML Command Examples

```
-- Select Query to retrieve high-scoring students
SELECT RollNo, StudentName, Marks
FROM tbl_Students
WHERE Marks >= 90
ORDER BY Marks DESC;

-- Insert Query to add a new record
INSERT INTO tbl_Students (RollNo, StudentName, Marks)
VALUES (101, 'Aarav Sharma', 94);

-- Update Query to adjust student fees
UPDATE tbl_Students
SET FeeStatus = 'Paid'
WHERE RollNo = 101;

-- Delete Query to remove inactive records
DELETE FROM tbl_Students
WHERE Status = 'Inactive';
```

5 Creating Forms and Reports

While Datasheet View offers quick record viewing, administrative database systems require presentable user interfaces for data collection and formal output summaries.

5.1 Forms: Concept and Benefits

A **Form** is a database object that provides a customized, user-friendly graphical interface containing visual controls (text boxes, drop-downs, buttons) for viewing, entering, editing, and navigating table records.

Important Note

Why Use Forms Instead of Datasheet View?

- Prevents accidental corruption or editing of adjacent table rows.
- Allows formatting record entry one record at a time with clear label prompts.
- Enables non-technical users to operate the database safely.

5.2 Creating Forms Using Form Wizard

The **Form Wizard** offers an 8-step creation path:

1. In Base, click **Forms** → **Use Wizard to Create Form...**
2. **Step 1: Field Selection** - Select the source table or query and choose fields to place on the form.
3. **Step 2: Setup a Subform** - (Optional) Check "Add Subform" to display related records from a child table.
4. **Step 3: Add Subform Fields** - Select fields for the subform.
5. **Step 4: Get Joined Fields** - Select matching join keys between main form and subform.
6. **Step 5: Arrange Controls** - Choose visual alignment (*Columnar - Labels Left*, *Columnar - Labels Top*, *As Data Sheet*, *In Blocks - Labels Left*).
7. **Step 6: Set Data Entry** - Specify form behavior (e.g., "The form is to display all data" or "Do not allow modification of existing data").
8. **Step 7: Apply Styles** - Choose color background palettes (e.g., *Blue*, *Gray*, *Water*) and field border styles (*3D look* or *Flat*).
9. **Step 8: Set Name** - Enter a form title and click **Finish**.

5.3 Form Controls and Properties

When opening a Form in **Design View**, users can insert and configure graphical interactive controls using the **Form Controls Toolbar**:

Table 4: Interactive Form Controls in LibreOffice Base

Control Name	UI Representation	Primary Purpose
Label Field	Static Text Box	Displays descriptive text prompts or instructions.
Text Box	Entry Field	Accepts or displays single-line or multi-line text/numbers.
List Box	Drop-down List	Displays a drop-down menu of choices loaded from a table.
Combo Box	Edit Combo List	Drop-down list that also allows typing custom entries.
Check Box	Square Check Box	Captures binary Yes/No or True/False boolean flags.
Option / Radio Button	Round Radio Button	Allows selecting exactly one option from a mutually exclusive group.
Push Button	Clickable Button	Triggers actions (e.g., Save Record, Reset, Next Record, Submit).

5.4 Reports: Concept and Features

A **Report** is a database object designed to compile, format, summarize, and present information extracted from tables or queries in an organized, printable layout for analysis and distribution.

Table 5: Comparative Analysis: Forms vs. Reports

Feature	Form	Report
Primary Purpose	Data entry, editing, and record navigation.	Data presentation, summarizing, and printing.
Interactivity	Fully interactive (read/write).	Static document view (read-only).
Direction	Two-way input/output interface.	One-way output presentation.
Grouping	Displays individual records or sub-forms.	Groups data with headers, footers, and grand totals.

5.5 Creating Reports Using Report Wizard

To build a formal report:

1. In Base, click **Reports** → **Use Wizard to Create Report....**
2. **Step 1: Field Selection** - Choose table/query and select fields to include.
3. **Step 2: Labeling Fields** - Rename display labels for report headers.
4. **Step 3: Grouping** - Add grouping levels (e.g., group students by Class or orders by City).

5. **Step 4: Sort Options** - Specify sorting hierarchy (up to 4 fields).
6. **Step 5: Choose Layout** - Select layout arrangement (*Columnar* or *Tabular*) and page orientation (*Portrait* or *Landscape*).
7. **Step 6: Create Report** - Specify report title, choose **Static Report** (fixed snapshot) or **Dynamic Report** (updates when underlying data changes), and click **Finish**.

6 Comprehensive Question Bank

6.1 Multiple Choice Questions (MCQs)

1. Which of the following software applications is an Open-Source Relational Database Management System?

(A) MS Access	(C) Oracle Database
(B) LibreOffice Base	(D) Microsoft SQL Server

Answer: (B) LibreOffice Base

Explanation: LibreOffice Base is a free, open-source RDBMS application distributed under the GNU LGPL license, whereas MS Access and Oracle are proprietary commercial software.

2. What is the default file extension for a database file created in LibreOffice Base?

(A) .odt	(C) .odb
(B) .ods	(D) .odp

Answer: (C) .odb

Explanation: LibreOffice Base saves database files using the OpenDocument Database extension (.odb). .odt is for Writer, .ods is for Calc, and .odp is for Impress.

3. In a relational table, a horizontal entity representing a complete set of attributes for one item is called a:

(A) Field	(C) Domain
(B) Tuple / Record	(D) Attribute

Answer: (B) Tuple / Record

Explanation: A horizontal row in a table is called a Tuple or Record, while a vertical column is called a Field or Attribute.

4. Which database key uniquely identifies each record in a table and can NEVER contain NULL values?

(A) Foreign Key	(C) Primary Key
(B) Candidate Key	(D) Alternate Key

Answer: (C) Primary Key

Explanation: A Primary Key enforces entity integrity by guaranteeing unique non-null identification for every row in a relation.

5. Which rule ensures that foreign key values in a child table must match a valid primary key value in the parent table?

(A) Entity Integrity	(C) Domain Integrity
(B) Referential Integrity	(D) Operational Integrity

Answer: (B) Referential Integrity

Explanation: Referential Integrity guarantees logical consistency between primary key values of a parent table and matching foreign keys in child tables.

6. Which data type in LibreOffice Base is used to store variable-length character text strings?

(A) CHAR	(C) INTEGER
(B) VARCHAR	(D) BOOLEAN

Answer: (B) VARCHAR

Explanation: VARCHAR (Variable Character) dynamically adjusts storage space to match the actual length of typed text up to the specified limit.

7. What type of relationship exists when one record in Table A is associated with multiple records in Table B, but a record in Table B links to only one record in Table A?

(A) One-to-One	(C) Many-to-Many
(B) One-to-Many	(D) Self-Referential

Answer: (B) One-to-Many

Explanation: A 1:N relationship connects a single parent record to multiple child records (e.g., one customer having multiple orders).

8. Which query design grid row determines whether a field is displayed in the final output result sheet?

(A) Alias	(C) Visible
(B) Sort	(D) Criterion

Answer: (C) Visible

Explanation: The **Visible** check-box row controls whether a field appears in the query results view.

9. Which SQL command is used to retrieve specific records from one or more tables?

(A) UPDATE	(C) SELECT
(B) INSERT	(D) DELETE

Answer: (C) SELECT

Explanation: The SELECT statement is the foundational Data Manipulation Language (DML) command used to query and fetch records.

10. Which database object is specifically designed to present compiled database insights in a formatted, printable layout?

(A) Form

(C) Query

(B) Table

(D) Report

Answer: (D) Report

Explanation: Reports compile and format structured output data into professional static documents tailored for printing or administrative presentation.

6.2 Assertion-Reasoning Questions

In the following questions, two statements are given: **Assertion (A)** and **Reason (R)**. Select the correct option:

- (A) Both A and R are true, and R is the correct explanation of A.
- (B) Both A and R are true, but R is NOT the correct explanation of A.
- (C) A is true, but R is false.
- (D) A is false, but R is true.

1. **Assertion (A):** A Database Management System eliminates data redundancy across operational files.

Reason (R): DBMS stores all operational data in a centralized repository accessible by authorized users.

Answer: (A) Both A and R are true, and R is the correct explanation of A.

Explanation: Centralized data storage prevents duplicate file copies from being maintained across separate departments, thereby eliminating redundancy.

2. **Assertion (A):** A Primary Key column in a relational table can contain NULL values if all other entries are unique.

Reason (R): Primary Keys enforce entity integrity to uniquely identify every record in a relation.

Answer: (D) A is false, but R is true.

Explanation: A Primary Key can NEVER contain NULL values under any circumstances because NULL signifies an unknown entity, violating entity integrity.

3. **Assertion (A):** Referencing a primary key in a child table creates a foreign key relationship.

Reason (R): Foreign keys enforce referential integrity between related relational tables.

Answer: (A) Both A and R are true, and R is the correct explanation of A.

Explanation: Foreign key attributes link child rows to parent primary keys, ensuring referential integrity rule compliance.

4. **Assertion (A):** Forms are static read-only database objects used exclusively for printing hard copies.
Reason (R): Reports present summarized database output formatted for formal presentation.
Answer: (D) A is false, but R is true.
Explanation: Assertion A is false because Forms are interactive two-way graphical screens used for data entry and record editing; Reports are the static printing objects.
5. **Assertion (A):** The wildcard criterion LIKE 'S%' matches any text string that begins with the letter 'S'.
Reason (R): In SQL pattern matching, the percentage symbol (%) represents zero, one, or multiple characters.
Answer: (A) Both A and R are true, and R is the correct explanation of A.
Explanation: The % wildcard matches any sequence of trailing characters, making 'S%' match string prefixes starting with 'S'.

6.3 One-Word / Brief Answer Questions

1. What is the term for raw, unorganized facts and figures?
Answer: Data.
2. Name the relational database model invented by Dr. E.F. Codd that uses 2D tables.
Answer: Relational Data Model.
3. Which key consists of a combination of two or more attributes to uniquely identify a record?
Answer: Composite Primary Key.
4. What is the horizontal row in a relational table called?
Answer: Tuple (or Record).
5. Which data type in LibreOffice Base is used to store boolean True/False flags?
Answer: BOOLEAN (or Yes/No).
6. Name the menu option in LibreOffice Base used to open the Relationships window.
Answer: Tools -> Relationships...
7. Which database object acts as an interactive graphical front-end for data entry?
Answer: Form.
8. Which grid row in Query Design View specifies filtering conditions?
Answer: Criterion.
9. Write the SQL aggregate function used to calculate the average value of a numeric field.
Answer: AVG().
10. What is the default delete rule option in referential integrity that prevents parent record deletion if child records exist?
Answer: No Action (or Restrict).

6.4 Differentiate Between Questions

1. Differentiate between Data and Information.

Feature	Data	Information
Processing	Raw, unprocessed facts.	Processed, organized data.
Context	Lacks context and meaning.	Contextualized with clear meaning.
Dependency	Independent foundational input.	Dependent directly on data.

2. Differentiate between Primary Key and Foreign Key.

Feature	Primary Key	Foreign Key
Uniqueness	Must contain unique values only.	Can contain duplicate values.
Null Values	Cannot contain NULL values.	Can accept NULL values (if allowed).
Role	Uniquely identifies records in parent table.	References primary key of parent table.

3. Differentiate between CHAR and VARCHAR Data Types.

Feature	CHAR	VARCHAR
Length	Fixed length string.	Variable length string.
Padding	Pads remaining spaces with blanks.	No padding added; allocates actual text length.
Storage	Constant memory allocation.	Dynamic memory allocation.

4. Differentiate between Forms and Reports.

Feature	Form	Report
Purpose	Interactive data entry and record editing.	Static formatted presentation for printing.
Interactivity	Editable (Read / Write).	Read-Only output document.
Focus	Operates primarily record by record.	Operates on summarized groups and totals.

5. Differentiate between One-to-Many and Many-to-Many Relationships.

Feature	One-to-Many (1:N)	Many-to-Many (N:M)
Linkage	One parent record links to multiple child records.	Multiple records link to multiple records.
Implementation	Direct foreign key in child table.	Requires an intermediate junction table.
Example	Customer to Orders.	Students to Courses.

6.5 Short Answer Questions

1. **Explain the concept of Data Redundancy and state how a DBMS controls it.**

Answer: Data redundancy refers to unnecessary duplication of identical data across multiple operational files. A DBMS controls redundancy by integrating operational files into a single centralized database. Information (such as a student address) is stored in one primary table, allowing all departments to access the single master record.

2. **What is Referential Integrity? Why is it important in RDBMS?**

Answer: Referential Integrity is a database integrity rule ensuring that foreign key values in a child table correspond to a valid, existing primary key in the referenced parent table. It is crucial because it prevents orphaned records and maintains relational consistency when parent records are updated or deleted.

3. **Briefly explain the two main panes of the Query Design Window in LibreOffice Base.**

Answer: The Query Design Window contains two main panes: 1. *Upper Pane (Tables Pane)*: Displays target table diagrams, field lists, and relationship connector lines. 2. *Lower Pane (Criteria Grid)*: Contains configurable rows (Field, Alias, Table, Sort, Visible, Criterion, Or) used to define fields, sorting order, and filtering conditions.

4. **List any four interactive Form Controls available in LibreOffice Base and state their usage.**

Answer: 1. *Text Box*: Accepts or displays text and numeric inputs. 2. *List Box*: Offers a drop-down menu of choices fetched from a database column. 3. *Check Box*: Toggles binary boolean Yes/No choices. 4. *Push Button*: Triggers operational actions such as saving, resetting, or submitting form records.

5. **What is the difference between Update Cascade and Delete Cascade in referential integrity rules?**

Answer: *Update Cascade* automatically updates foreign key values in matching child rows whenever the primary key value of a parent record is modified. *Delete Cascade* automatically deletes all matching child rows from the child table whenever the parent record is deleted.

6.6 Long Answer Questions

1. **Explain the major advantages of using a Relational Database Management System (RDBMS) over traditional paper or file-based record keeping systems.**

Answer: Relational Database Management Systems provide comprehensive architectural advantages over traditional file systems:

- **Minimal Data Redundancy:** Traditional file systems store redundant student or employee details in separate files for accounts, library, and administration. An RDBMS stores data in a centralized location, keeping duplication to a minimum.
- **Enforced Data Consistency:** Controlling redundancy ensures data consistency. Updating a student's contact details once in the central table reflects across all system modules immediately.
- **Controlled Data Access and Security:** RDBMS platforms enforce user authentication and role-based permissions, restricting sensitive fields (e.g., marks, salary) to authorized personnel.

- **Data Integrity Constraints:** RDBMS software enforces rules (e.g., Primary Key uniqueness, range limits, data type boundaries) to prevent erroneous data entry.
 - **Automated Backup and Recovery:** Built-in transaction logging automatically restores lost records in the event of system or hardware failure.
2. **Describe step-by-step how to create a table in Design View in LibreOffice Base, including setting a Primary Key and configuring field properties.**

Answer: Creating a table in Design View follows these structured steps:

- **Open Design View:** Launch LibreOffice Base, click **Tables** in the Database Objects Pane, and click **Create Table in Design View....**
 - **Define Field Names and Data Types:** In the upper grid pane, enter field names in the **Field Name** column (e.g., StudentID, Name, DOB, FeePaid) and select appropriate data types from the **Field Type** drop-down menu (INTEGER, VARCHAR, DATE, DECIMAL).
 - **Configure Field Properties:** Click each field row to adjust its settings in the lower pane (e.g., set **Length** for text fields, or set **Entry required** to **Yes**).
 - **Assign Primary Key:** Right-click the gray row selector icon beside the designated field (e.g., StudentID) and choose **Primary Key**. A key icon will mark the field.
 - **Save Table:** Click **Save** on the toolbar, type a table name (e.g., tbl_StudentDetails), and click **OK**.
3. **Explain the process of setting up a Query in Design View to filter records matching specific numeric and text conditions, including sorting and calculations.**

Answer: Constructing a query in Design View involves the following process:

- **Launch Query Design View:** Click **Queries** in the left pane and select **Create Query in Design View....**
- **Add Target Tables:** In the **Add Table or Query** dialog box, select the source tables (e.g., tbl_Students) and click **Add**.
- **Populate Grid Fields:** Double-click desired field names (RollNo, StudentName, City, Marks) to add them into the lower Criteria Grid columns.
- **Set Sorting Order:** In the **Sort** row under Marks, select **Descending** to rank high scores at the top.
- **Apply Condition Criteria:** Set Marks ≥ 80 and City = 'Delhi'.
- **Run and Save Query:** Click the **Run Query** icon (F5) on the toolbar to preview output. Click **Save**, enter a query name (e.g., qry_Delhi_Toppers), and click **OK**.

6.7 Case-Study / Competency-Based Questions

1. Case Study 1: School Sports Day Management System

Scenario: Sunita, the Sports Coordinator at OAV Bibina, is organizing the Annual Sports Meet. She manages data for 500 participating students, 12 sports events (e.g., 100m race, Long Jump, Chess), and event result scores. Initially, she maintained a single spreadsheet file, but noticed duplicate student entries, invalid event IDs, and conflicting records when multiple event heads modified the sheet. She decides to migrate the system to a relational database in LibreOffice Base consisting of two tables: tbl_Students (StudentID, StudentName, Class, House) and tbl_Events (EventID, EventName, StudentID, Position).

- (a) Which field should be designated as the Primary Key in tbl_Students?
Answer: StudentID should be the Primary Key because it uniquely identifies every student.
- (b) What type of relationship exists between tbl_Students and tbl_Events?
Answer: A One-to-Many (1:N) relationship (One student can participate in multiple events).
- (c) Which database integrity rule will prevent Sunita from entering an event result for a StudentID that does not exist in tbl_Students?
Answer: Referential Integrity.

2. Case Study 2: Retail Supermarket Billing and Inventory

Scenario: Raj operates a local supermarket and uses LibreOffice Base to manage sales. His database contains tbl_Products (ProductID, ProductName, Category, UnitPrice, StockQty) and tbl_Sales (SaleID, ProductID, QtySold, SaleDate). Raj wants to generate automatic daily sales reports, restrict billing staff from directly modifying table schemas, and quickly search products with stock lower than 10 units.

- (a) Which database object should Raj create to provide billing staff a user-friendly data entry screen that prevents direct table access?
Answer: A Form.
- (b) Write the SQL query criterion Raj should enter in the **Criterion** row under StockQty in Query Design View to identify low-stock items.
Answer: < 10
- (c) Which database object should Raj use to print formatted daily sales summaries for his accounting team?
Answer: A Report.

3. Case Study 3: Central Library Automation Project

Scenario: An Academic Library is automating its book issue register using LibreOffice Base. The system maintains two tables: tbl_Books (BookID, Title, Author, Category) and tbl_Issue (IssueID, BookID, MemberID, IssueDate, ReturnDate).

- (a) Identify the Foreign Key field in tbl_Issue.
Answer: BookID (references the Primary Key of tbl_Books).
- (b) If a book is damaged and permanently deleted from tbl_Books, which referential integrity delete rule ensures that all past issue records for that book are also deleted automatically?
Answer: Delete Cascade.
- (c) Which Base data type should be selected for the IssueDate field in tbl_Issue?
Answer: DATE.